

Generalized Conjugate Directions

J.E. Dennis, Jr.<sup>1</sup>

and

Kathryn Turner<sup>2</sup>

Technical Report 85-11, December 1985.

Revised, May 1986

---

<sup>1</sup>Mathematical Sciences Department, Rice University, Houston, Texas 77251-1892. Research sponsored by NSF MCS81-16779, DOE DE-AS05-82ER13016, ARO DAAG-29-83-K-0035.

<sup>2</sup>Mathematical Sciences Department, Rice University, Houston, Texas 77251-1892. Research sponsored by ARO DAAG-29-83-K-0035.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>MAY 1986</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-1986 to 00-00-1986</b>	
4. TITLE AND SUBTITLE <b>Generalized Conjugate Directions</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>26</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

## *Abstract*

This paper presents a simple unifying framework for a wide class of conjugate directions algorithms whose iterates minimize some quadratic functional over a subspace. Our approach is motivated by its advantages for nonlinear minimization, but the purpose of this paper is to present the greatly simplified convergence analysis that results for the linear case.

## *Key words*

Generalized conjugate directions methods, conjugate gradients, conjugate residuals, Krylov subspaces, iterative methods.



## GENERALIZED CONJUGATE DIRECTIONS

### Introduction

We present a simple unifying framework for a class of conjugate directions algorithms for the solution of the linear system

$$Ax = b, \text{ where } A \in \mathbb{R}^{n \times n} \text{ is large, sparse, and nonsingular.}$$

The class under consideration consists of algorithms that minimize an error functional over a subspace or affine space at each step. This class includes the standard and the preconditioned conjugate gradient algorithms, the conjugate residual algorithm, Craig's method, Elman's generalized conjugate residual algorithm, Vinsome's ORTHOMIN algorithm, the ORTHODIR algorithm of Young and Jea, the GMRES algorithm of Saad and Schultz, and the truncated algorithms ORTHOMIN( $k$ ) and ORTHODIR( $s$ ). The main point of this paper is that our approach leads to a single simple geometric theorem giving the standard convergence results for all the untruncated methods without the usual clutter of lemmas. Another theorem, which we give without proof because the proof is so similar, suffices for the truncated methods. Furthermore, this paper should help clear up the common misconceptions about the relationship between these methods and the Krylov subspace.

The approach we have taken in formulating the Generalized Conjugate Directions (GCD) algorithm is based on an idea suggested to us by Peter Huber, who reported finding it useful in practice for nonlinear problems. It is implicit in work by others as well, especially Miele and Cantrell (1969), Cantrell (1969), Cragg and Levy (1969), and Nazareth (1984). The idea is simple; its key feature is to take the inverse point of view to the usual one of generating conjugate directions and then minimizing in the last direction generated.

Our formulation is similar in spirit to Axelsson's generalized conjugate gradient algorithm, but there is an important difference: we have used a basis for the subspace over which minimization must be carried out at each step that allows us to explicitly solve the minimization problem

for the general case. Our formulation facilitates analysis of domains of convergence and the number of previous directions required to be saved for these algorithms. We show a q-linear convergence rate under a very mild hypothesis for the class of algorithms.

The conjugate directions formulation we use for solving a linear system is also applicable to the solution by conjugate directions of equality constrained quadratic programming problems, and we discuss this adaptation of the GCD algorithm.

### The Generalized Conjugate Directions Algorithm

Our GCD algorithm is stated as a means of finding the minimizer of the quadratic

$$q(x) = \frac{1}{2} x^T H x - h^T x ,$$

where we assume  $H$  is symmetric positive definite. Thus,  $q(x)$  has a unique minimizer, and finding it is equivalent to finding the zero of

$$\nabla q(x) = Hx - h .$$

To apply the algorithm to the solution of  $Ax = b$ , one may use  $H = A$ ,  $h = b$  if  $A$  is symmetric positive definite, or  $H = ZA$ ,  $h = Zb$ , where  $Z$  is nonsingular and  $ZA$  is symmetric positive definite.

We use the notation

$$\bar{r}_k = -\nabla q(x_k) = h - Hx_k$$

and

$$r_k = b - Ax_k .$$

The minimizer of  $q(x)$  is

$$x^* = H^{-1}h ,$$

and the minimum value of  $q(x)$  is

$$q(x^*) = -\frac{1}{2} h^T H^{-1} h .$$

We define  $\|x - x^*\|_H^2 \equiv (x - x^*)^T H(x - x^*)$ , and note that

$$(x - x^*)^T H(x - x^*) = x^T Hx - 2x^T Hx^* + x^{*T} Hx^* = 2[q(x) - q(x^*)],$$

so that minimizing  $q(x)$  is equivalent to minimizing  $\|x - x^*\|_H$ .

The GCD algorithm produces a sequence  $p_1, p_2, \dots$  of linearly independent, mutually  $H$ -conjugate ( $p_i^T H p_j = 0, i \neq j$ ) directions. Conceptually, we do not choose these directions. Rather, at each step of the algorithm, we choose a direction  $d_k$  in any manner whatever, requiring only that it not be orthogonal to the gradient of  $q$  at  $x_{k-1}$ . We determine the next iterate  $x_k$  as the minimizer of  $q(x)$  in  $sp\{p_1, \dots, p_{k-1}, d_k\}$ , the subspace spanned by  $\{p_1, \dots, p_{k-1}, d_k\}$ , and define  $p_k \equiv x_k - x_{k-1}$ . Clearly, the choice of  $d_k$ , which we now discard, determines  $p_k$ .

For ease of exposition, we use the initial guess  $x_0 = 0$ . This is not restrictive. If a better approximate solution, say  $\hat{x}$ , is known, the problem  $Ax = b$  with initial guess  $\hat{x}$  can be solved by solving  $A(x - \hat{x}) = b - A\hat{x}$  with initial guess 0. It will be useful to let  $\underset{x \in S}{\operatorname{argmin}} q(x)$  denote the minimizer of  $q(x)$  on  $S$ . Our GCD algorithm is as follows.

*Generalized Conjugate Directions Algorithm*

$$x_0 = 0, \quad \bar{r}_0 = h, \quad k = 1.$$

while  $\bar{r}_{k-1} \neq 0$

do

    get  $d_k$  such that  $d_k^T \bar{r}_{k-1} \neq 0$

$$x_k = \underset{x \in sp\{p_1, \dots, p_{k-1}, d_k\}}{\operatorname{argmin}} q(x)$$

$$p_k = x_k - x_{k-1}$$

$$\bar{r}_k = \bar{r}_{k-1} - H p_k$$

$$k = k + 1$$

end do

The GCD algorithm with choices of  $H$ ,  $h$ , and  $d_k$  given in columns 3-5 in the following table is equivalent to the algorithms named in column 1.

Algorithm	Assumptions	$H$	$h$	$d_k$
CG	$A$ spd	$A$	$b$	$r_{k-1}$
CG applied to normal equations	$A$ nonsingular	$A^T A$	$A^T b$	$\bar{r}_{k-1}$
Craig's Method	$A$ nonsingular	$AA^T$	$b$	$r_{k-1}$
CR	$A$ spd	$A^T A$	$A^T b$	$r_{k-1}$
PCG	$A$ spd	$A$	$b$	$M^{-1}r_{k-1}$
GCR	$A + A^T$ pd	$A^T A$	$A^T b$	$r_{k-1}$
ORTHOMIN	$A + A^T$ pd	$A^T A$	$A^T b$	$r_{k-1}$
ORTHODIR	use $Z$ such that $Z + Z^T$ is pd and $ZA$ is spd	$ZA$	$Zb$	$d_1 = b$ ; for $k \geq 2$ , $d_k = Ap_{k-1}$
GMRES	$A$ nonsingular	$A^T A$	$A^T b$	$\in \{v_1, \dots, v_m\}$

CG denotes the standard conjugate gradient algorithm, and CR denotes the conjugate residual algorithm. For the solution of the linear system  $Ax = b$  using Craig's method, put  $x = A^T y$  and solve  $AA^T y = b$ . The computation can be arranged so that iterates in  $y$  are not actually computed. PCG denotes the preconditioned conjugate gradient algorithm. To solve the linear system  $Ax = b$  using the preconditioned conjugate gradient algorithm, one finds a nonsingular symmetric matrix  $C^{-1}$  such that  $C^{-1}AC^{-1}$  is better conditioned than  $A$ , and transforms  $Ax = b$  to  $(C^{-1}AC^{-1})(Cx) = C^{-1}b$ . The matrix  $M$  that appears in the choice of  $d_k$  for PCG is defined to be  $C^2$ , and is called the preconditioner. Vinsome's ORTHOMIN using all previous directions is equivalent to Elman's generalized conjugate residual algorithm, denoted GCR. The ORTHODIR algorithm as given by Young and Jea is stated as requiring only that  $Z$  be nonsingular and  $(ZA) + (ZA)^T$  positive definite, but ORTHODIR is included in the class of algorithms under present consideration (those algorithms that minimize the error functional  $\|x - x^*\|_H$  over a subspace or affine space at each step) only if  $ZA$  is symmetric positive definite. Young and Jea



require the symmetric part of  $Z$  positive definite in order to ensure  $x_k \neq x_{k-1}$ . We have included this condition here, and will first discuss the equivalence of ORTHODIR and GCD when this condition is satisfied. Later, we show a way of viewing the ORTHODIR algorithm in our framework when  $Z$  is just nonsingular. The GMRES algorithm of Saad and Schultz does not compute iterates  $x_k$  to approximate  $x^*$ . Instead, it builds an orthonormal basis  $\{v_1, \dots, v_m\}$  for the Krylov space  $K(b, A, n) = \text{sp}\{b, Ab, \dots, A^{n-1}b\}$  and, using this basis, determines the solution to  $Ax = b$  as the minimizer of  $\|Ax - b\|_2$  over  $K(b, A, n)$ . The vectors  $\{v_i\}$  are determined by Arnoldi's method:

$$v_1 = \frac{b}{\|b\|};$$

for  $j = 1, \dots, m-1$

$$\begin{aligned} \hat{v}_{j+1} &= Av_j - \sum_{i=1}^j v_i^T Av_j v_i \\ v_{j+1} &= \frac{\hat{v}_{j+1}}{\|\hat{v}_{j+1}\|}. \end{aligned}$$

The truncated algorithms ORTHOMIN( $k$ ) and ORTHODIR( $s$ ) are equivalent to GCD( $m$ ), which we define later, with the same choices of  $H$ ,  $h$ , and  $d_k$  shown for ORTHOMIN and ORTHODIR, respectively.

Here is the first of our two main results. Notice that the Krylov subspace is totally separated from the convergence analysis for the general method. Of course, a choice of  $d_k$  that implies minimization on a Krylov subspace may greatly reduce the computational complexity of the associated method, as we shall show. Furthermore, the Krylov subspace is essential to the Chebyshev polynomial error analysis usually associated with such methods.

*Theorem 1:* If  $H$  is symmetric positive definite, then the following statements are true about the GCD algorithm:

- (i) The algorithm terminates after no more than  $n$  steps, and terminates if and only if  $x_{k-1} = x^*$ .

- (ii)  $x_k$  minimizes  $q(x)$  on  $sp\{p_1, \dots, p_k\}$ .
- (iii) Every  $p_k$  generated by the algorithm is nonzero.
- (iv)  $\bar{r}_k^T p_j = 0, 1 \leq j \leq k$
- (v)  $p_i^T H p_j = 0, i \neq j$
- (vi)  $\dim sp\{p_1, \dots, p_k\} = k$
- (vii)  $p_i^T H p_i = p_i^T h$  for all  $i$

If in addition  $d_k \in sp\{d_1, Bp_1, \dots, Bp_{k-1}\}$  for some matrix  $B$ , then

- (viii)  $sp\{p_1, \dots, p_k\} = K(d_1, B, k)$ , so that  $x_k$  minimizes  $q(x)$  on  $K(d_1, B, k)$ .
- (ix)  $\bar{r}_k^T B p_i = 0, 1 \leq i \leq k-1$ .

*Proof:*

The algorithm can terminate only because some  $\bar{r}_{k-1} = 0$ . If  $0 = \bar{r}_{k-1} = -\nabla q(x_{k-1})$ , then  $x_{k-1}$  is a critical point for  $q$ . Since  $\nabla^2 q(x_{k-1}) = H$  is positive definite, this is necessary and sufficient for  $x_{k-1} = x^*$ . We shall prove that the algorithm terminates after not more than  $n$  steps by proving (ii) and (vi) for  $k = n$ , because then by (vi),  $\mathbf{R}^* = sp\{p_1, \dots, p_k\}$  and  $x_n = x^*$  by (ii).

Let us proceed by induction. For  $k = 1$ , if  $\bar{r}_0 (= h) = 0$ , we are finished. Otherwise, we choose  $d_1$  such that  $d_1^T \bar{r}_0 \neq 0$ . We can certainly do this by taking  $d_1 = \bar{r}_0$ , for example. Since  $d_1^T \bar{r}_0 \neq 0$ ,  $sp\{d_1\}$  contains a descent direction for  $q$  from  $x_0$ . Thus,

$$p_1 = x_1 - x_0 \neq 0,$$

which anchors the induction for (iii) and (vi), and

$$sp\{p_1\} = sp\{d_1\},$$

which anchors (ii). Since  $x_1$  minimizes  $q$  on  $sp\{p_1\}$ ,  $sp\{p_1\}$  cannot contain a descent direction for  $q$  from  $x_1$ . Thus

$$0 = -\nabla q(x_1)^T p_1 = \bar{r}_1^T p_1,$$

anchoring (iv), and this is

$$= (h - Hx_1)^T p_1 = (h - Hp_1)^T p_1,$$

which anchors (vii). Also,

$$sp\{p_1\} = sp\{d_1\} = K(d_1, B, 1)$$

for any  $B$ , which anchors (viii). Statements (v) and (ix) are vacuously true.

Now assume that (i)-(vii) hold for  $1 \leq j \leq k-1$ . We have already taken care of the case when the algorithm terminates because  $\bar{r}_{k-1} = 0$ , so we assume  $\bar{r}_{k-1} \neq 0$ . Then we can choose  $d_k$ , and  $sp\{d_k\}$  contains a descent direction for  $q$  from  $x_{k-1}$ . Thus

$$0 \neq x_k - x_{k-1} = p_k, \quad (\text{iii}).$$

Since  $x_{k-1} \in sp\{p_1, \dots, p_{k-1}\}$ , and  $x_k \in sp\{p_1, \dots, p_{k-1}, d_k\}$ , we have  $d_k \in sp\{p_1, \dots, p_k\}$ . By the induction hypothesis,  $\bar{r}_{k-1}^T p_j = 0$  for  $1 \leq j \leq k-1$ , so  $d_k \notin sp\{p_1, \dots, p_{k-1}\}$ . Hence,

$$sp\{p_1, \dots, p_k\} = sp\{p_1, \dots, p_{k-1}, d_k\}, \quad (\text{ii})$$

and

$$\begin{aligned} \dim sp\{p_1, \dots, p_k\} &= \dim sp\{p_1, \dots, p_{k-1}, d_k\} \\ &= \dim sp\{p_1, \dots, p_{k-1}\} + 1 \\ &= k-1+1 = k, \quad (\text{vi}). \end{aligned}$$

Since  $x_k$  minimizes  $q$  on  $sp\{p_1, \dots, p_k\}$ ,

$$0 = \nabla q(x_k)^T p_i = -\bar{r}_k^T p_i \text{ for } 1 \leq i \leq k, \quad (\text{iv}).$$

Note that  $x_k = \sum_{j=1}^k p_j$ , so that  $\bar{r}_k = h - \sum_{j=1}^k H p_j$ . Thus for  $i < k$ ,

$$0 = p_i^T \bar{r}_k = p_i^T h - \sum_{j=1}^k p_i^T H p_j.$$

By the induction hypothesis, this is

$$\begin{aligned} &= p_i^T h - p_i^T H p_i - p_i^T H p_k \\ &= -p_i^T H p_k, \quad (\text{v}). \end{aligned}$$

Furthermore,

$$0 = p_k^T \bar{r}_k = p_k^T h - \sum_{j=1}^k p_k^T H p_j = p_k^T h - p_k^T H p_k, \quad (\text{vii}).$$

Now assume (viii) holds for  $1 \leq j \leq k-1$ , and that  $d_k \in sp\{d_1, B p_1, \dots, B p_{k-1}\}$ . There exist  $\alpha$  and  $\beta_j$ ,  $1 \leq j \leq k-1$ , such that

$$d_k = \alpha d_1 + \sum_{j=1}^{k-1} \beta_j B p_j = \left( \alpha d_1 + \sum_{j=1}^{k-2} \beta_j B p_j \right) + \beta_{k-1} B p_{k-1}.$$

By the induction hypothesis, for  $1 \leq i \leq k-1$ ,  $p_i \in K(d_1, B, k-1)$ , so that the first term  $\in K(d_1, B, k-1)$  and the second term  $\in K(d_1, B, k)$ ; thus  $d_k \in K(d_1, B, k)$ . This, together with  $sp\{p_1, \dots, p_k\} = sp\{p_1, \dots, p_{k-1}, d_k\}$ , gives

$$sp\{p_1, \dots, p_k\} = K(d_1, B, k), \quad (\text{viii}).$$

For (ix), observe that by (iv) and (viii),  $\bar{r}_k$  is orthogonal to  $K(d_1, B, k)$ . For  $i \leq k-1$ ,  $p_i \in K(d_1, B, k-1)$ , so that  $B p_i \in K(d_1, B, k)$ . •

At each step of the algorithm,  $x_k$  is to be determined as

$$x_k = \underset{s \in sp\{p_1, \dots, p_{k-1}, d_k\}}{\operatorname{argmin}} q(x).$$

Let  $P_k = [p_1, \dots, p_{k-1}, d_k]$ ,  $x = P_k c$ , and

$$\hat{q}(c) = q(P_k c) = \frac{1}{2} c^T (P_k^T H P_k) c - h^T P_k c.$$

Then  $\nabla \hat{q}(c) = P_k^T H P_k c - P_k^T h$ . Since  $H$  is positive definite and  $P_k$  has full column rank,  $P_k^T H P_k$  is positive definite. Hence,  $x_k$  can be found by solving the linear system  $P_k^T H P_k c = P_k^T h$  for  $c$ , and setting  $x_k = P_k c$ .

The matrix of the system of linear equations that determines  $x_k$  and hence  $p_k$  is symmetric with an arrowhead structure in the last row and column by part (v) of Theorem 1. All the standard recursions for  $p_k$  in each algorithm follow in an insightful and simple way from the general solution of the system. The way some algorithms under some assumptions can be seen to need only  $p_{k-1}$  and  $d_k$  to generate  $p_k$  is that the last row and column are zero in all but the last two elements.

By (v) and (vii) of Theorem 1, the linear system  $P_k^T H P_k c = P_k^T h$  is

$$\begin{bmatrix} p_1^T H p_1 & & & d_k^T H p_1 \\ & p_2^T H p_2 & & d_k^T H p_2 \\ & & \ddots & \vdots \\ & & & p_{k-1}^T H p_{k-1} & d_k^T H p_{k-1} \\ d_k^T H p_1 & d_k^T H p_2 & \dots & d_k^T H p_{k-1} & d_k^T H d_k \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{k-1} \\ c_k \end{bmatrix} = \begin{bmatrix} p_1^T H p_1 \\ p_2^T H p_2 \\ \vdots \\ p_{k-1}^T H p_{k-1} \\ d_k^T h \end{bmatrix}$$

The solution to the linear system is

$$c = \begin{bmatrix} 1 - \alpha_k \frac{p_1^T H d_k}{p_1^T H p_1} \\ 1 - \alpha_k \frac{p_2^T H d_k}{p_2^T H p_2} \\ \vdots \\ 1 - \alpha_k \frac{p_{k-1}^T H d_k}{p_{k-1}^T H p_{k-1}} \\ \alpha_k \end{bmatrix}, \quad \text{where } \alpha_k = \frac{d_k^T h - \sum_{j=1}^{k-1} d_k^T H p_j}{d_k^T H d_k - \sum_{j=1}^{k-1} \frac{(d_k^T H p_j)^2}{p_j^T H p_j}}.$$

The numerator of  $\alpha_k$  is equal to  $d_k^T \bar{r}_{k-1}$ , and let us define, for  $1 \leq j \leq k-1$ ,

$$\beta_j^{(k)} = \frac{-d_k^T H p_j}{p_j^T H p_j}.$$

Then

$$\alpha_k = \frac{d_k^T \bar{r}_{k-1}}{d_k^T H (d_k + \sum_{j=1}^{k-1} \beta_j^{(k)} p_j)}$$

and

$$\begin{aligned} x_k &= P_k c = \sum_{j=1}^{k-1} p_j + \alpha_k (d_k + \sum_{j=1}^{k-1} \beta_j^{(k)} p_j) \\ &= x_{k-1} + \alpha_k (d_k + \sum_{j=1}^{k-1} \beta_j^{(k)} p_j). \end{aligned}$$

When  $d_k$  is chosen so that  $d_k^T H p_i = 0$  for  $1 \leq i \leq k-2$ , the solution has the shorter form

$$\beta_k \equiv \beta_{k-1}^{(k)} = \frac{-d_k^T H p_{k-1}}{p_{k-1}^T H p_{k-1}}, \quad \alpha_k = \frac{d_k^T \bar{r}_{k-1}}{d_k^T H (d_k + \beta_k p_{k-1})},$$

and

$$x_k = x_{k-1} + \alpha_k(d_k + \beta_k p_{k-1}).$$

The standard conjugate gradient algorithm, conjugate gradient algorithm applied to the normal equations, conjugate residual algorithm, preconditioned conjugate gradient algorithm, and Craig's method all choose  $d_k$  so that only  $p_{k-1}$  and  $d_k$  are needed to generate  $p_k$  and  $x_k$ .

For CG and CR,

$$d_k = r_{k-1} = b - Ax_{k-1} = b - \sum_{j=1}^{k-1} Ap_j,$$

so that the matrix  $B$  that appears in the additional hypothesis of Theorem 1 is  $A$ . For CG

$$d_k^T H p_i = r_{k-1}^T A p_i = \bar{r}_{k-1}^T A p_i,$$

and for CR

$$d_k^T H p_i = r_{k-1}^T A^T A p_i = \bar{r}_{k-1}^T A p_i,$$

since  $A$  is symmetric. Thus, for both of these algorithms,

$$d_k^T H p_i = 0 \quad \text{for } 1 \leq i \leq k-2,$$

by part (ix) of Theorem 1. For CG applied to the normal equations, which we henceforth refer to as CGNE,

$$d_k = \bar{r}_{k-1} = h - Hx_{k-1} = h - \sum_{j=1}^{k-1} H p_j,$$

and for Craig's method,

$$d_k = r_{k-1} = b - AA^T y_{k-1} = h - \sum_{j=1}^{k-1} H p_j,$$

so that the matrix  $B$  is  $H$  for each of these algorithms, and

$$d_k^T H p_i = \bar{r}_{k-1}^T H p_i = 0 \quad \text{for } 1 \leq i \leq k-2.$$

For PCG,

$$d_k = M^{-1} r_{k-1},$$

and  $B$  is  $M^{-1}A$ . Since  $M$  is symmetric,

$$d_k^T H p_i = (r_{k-1}^T M^{-1}) A p_i = \bar{r}_{k-1}^T (M^{-1} A) p_i = 0 \quad \text{for } 1 \leq i \leq k-2.$$

GCR (ORTHOMIN) corresponds to the same choices of  $H$ ,  $h$ , and  $d_k$  as CR, but since this algorithm is applied when  $A$  is not symmetric, all of the previous directions contribute to the calculation and must be saved.

For ORTHODIR,  $B$  is  $A$ , and

$$d_k^T H p_i = p_{k-1}^T A^T Z A p_i.$$

Young and Jea refer to choosing  $Z$  such that  $Z$  (as well as  $ZA$ ) is symmetric positive definite as the “symmetrizable” case. If  $Z$  is symmetric, the above is

$$= (\bar{r}_{k-2} - \bar{r}_{k-1})^T A p_i,$$

so that

$$d_k^T H p_i = 0 \quad \text{for } 1 \leq i \leq k-3,$$

and only the last two previous directions must be kept. If  $Z$  is not symmetric, all previous directions contribute to the determination of the iterates.

We now verify convergence of these algorithms. For convergence of GCD, we require only that  $d_k^T \bar{r}_{k-1} \neq 0$  whenever  $\bar{r}_{k-1} \neq 0$ . This condition is easily seen to be satisfied for CG, CGNE, Craig’s method, CR, PCG, GCR (ORTHOMIN), GMRES, and ORTHODIR, under the restrictions we have shown for the applicability of each algorithm:

CG	$d_k^T \bar{r}_{k-1} = r_{k-1}^T r_{k-1}$
CGNE	$d_k^T \bar{r}_{k-1} = \bar{r}_{k-1}^T \bar{r}_{k-1}$
Craig’s method	$d_k^T \bar{r}_{k-1} = r_{k-1}^T r_{k-1}$
CR	$d_k^T \bar{r}_{k-1} = r_{k-1}^T A r_{k-1}$
PCG	$d_k^T \bar{r}_{k-1} = r_{k-1}^T M^{-1} r_{k-1}$
GCR, ORTHOMIN	$d_k^T \bar{r}_{k-1} = r_{k-1}^T A r_{k-1} = r_{k-1}^T \left( \frac{A + A^T}{2} \right) r_{k-1}$

For GMRES, we cannot ensure convergence for general nonsingular  $A$  by selecting  $d_k = v_k$ , but we can guarantee this convergence by a slightly more judicious choice of  $d_k$ , as we shall discuss momentarily. The choice  $d_k = v_k$  gives

$$sp\{p_1, \dots, p_k\} = sp\{d_1, \dots, d_k\} = sp\{v_1, \dots, v_k\} = K(b, A, k)$$

and

$$\begin{aligned} \bar{r}_{k-1}^T d_k &= \frac{\bar{r}_{k-1}^T \left( A v_{k-1} - \sum_{i=1}^{k-1} v_i^T A v_{k-1} v_i \right)}{\|\hat{v}_k\|} \\ &= \frac{\bar{r}_{k-1}^T A v_{k-1}}{\|\hat{v}_k\|} \\ &= \frac{\bar{r}_{k-1}^T A \sum_{i=1}^{k-1} c_i p_i}{\|\hat{v}_k\|}, \text{ for some scalars } \{c_i\} \text{ with } c_{k-1} \neq 0 \\ &\quad \text{(because } v_{k-1} \notin sp\{p_1, \dots, p_{k-2}\}) \\ &= \frac{c_{k-1}}{\|\hat{v}_k\|} \bar{r}_{k-1}^T (r_{k-2} - r_{k-1}) \\ &= \frac{-c_{k-1}}{\|\hat{v}_k\|} \bar{r}_{k-1}^T r_{k-1} \\ &= \frac{-c_{k-1}}{\|\hat{v}_k\|} r_{k-1}^T A r_{k-1}. \end{aligned}$$

Hence, if the symmetric part of  $A$  is indefinite, we may have  $\bar{r}_{k-1}^T v_k = 0$  at some step before the minimizer has been found, so that  $v_k$  will not be a suitable choice for  $d_k$ . If this should occur, however, we may select  $d_k$  and subsequent directions  $d_{k+1}, \dots, d_m$  from  $\{v_k, \dots, v_m\}$  in an order that gives  $\bar{r}_{k-1}^T d_k \neq 0$  at each step, until  $x^*$  is found. If  $\bar{r}_{k-1}^T v_i = 0$  for all remaining  $i$ ,  $k \leq i \leq m$ , then  $x_{k-1}$  minimizes  $q(x)$  on  $K(b, A, n)$ , and we are finished. We have the latitude of using the direction vectors  $\{v_i\}$  in any order in an algorithm equivalent to GMRES, since the iterates  $\{x_i\}$  are not actually computed in the GMRES algorithm. The point here is that this allows our convergence analysis to apply to GMRES.



For ORTHODIR, we are assured of convergence using the easily computed choice of  $d_k$  shown in the table if the symmetric part of  $Z$  is positive definite (or negative definite). If the symmetric part of  $Z$  is indefinite, this choice of  $d_k$  will not ensure descent at each step. In that case, we can guarantee convergence by a strategy similar to that employed for GMRES. For ORTHODIR, using  $d_1 = b$  and  $d_k = Ap_{k-1}$  for  $k \geq 2$ , we have

$$d_1^T \bar{r}_0 = b^T Z b ,$$

and for  $k \geq 2$ ,

$$\begin{aligned} d_k^T \bar{r}_{k-1} &= \bar{r}_{k-1}^T A p_{k-1} \\ &= \bar{r}_{k-1}^T (r_{k-2} - r_{k-1}) . \end{aligned}$$

Note that since  $\bar{r}_{k-1}$  is orthogonal to  $K(b, A, k-1)$ ,

$$\bar{r}_{k-1}^T r_{k-2} = \bar{r}_{k-1}^T (b - A x_{k-2}) = \bar{r}_{k-1}^T (b - \sum_{i=1}^{k-2} A p_i)$$

so

$$\begin{aligned} d_k^T \bar{r}_{k-1} &= -\bar{r}_{k-1}^T r_{k-1} \\ &= -r_{k-1}^T Z r_{k-1} . \end{aligned}$$

Thus, if the symmetric part of  $Z$  is positive definite (or negative definite),  $d_k^T \bar{r}_{k-1}$  is always nonzero whenever  $\bar{r}_{k-1}$  is nonzero, and ORTHODIR converges. Now, if the symmetric part of  $Z$  is indefinite, ORTHODIR is not guaranteed to give descent at every step, but is still convergent. ORTHODIR generates a sequence of vectors  $\{q_i\}$  that are mutually  $ZA$ -conjugate (when the symmetric part of  $ZA$  is positive definite) and that form a basis for the Krylov space  $K(b, A, n)$ . These vectors are determined as

$$q_0 = r_0 = b$$

and for  $k = 1, \dots, t$

$$q_k = A q_{k-1} + \sum_{j=0}^{k-1} \hat{\beta}_j^{(k)} q_j ,$$

where  $t+1$  is the dimension of the Krylov space  $K(b, A, n)$ , and for  $0 \leq j \leq k-1$ ,

$$\hat{\beta}_j^{(k)} = - \frac{q_j^T Z A^2 q_{k-1}}{q_j^T Z A q_j}$$

The ORTHODIR algorithm using an auxiliary matrix  $Z$  that is only required to be nonsingular can be viewed as a GCD algorithm in the following way. The distinct iterates of ORTHODIR are produced by GCD with the choice  $d_k \in \{q_0, \dots, q_t\}$ , with  $d_k = q_{k-1}$  provided  $q_{k-1}^T \bar{r}_{k-1} \neq 0$ . If  $q_{k-1}^T \bar{r}_{k-1} = 0$ , then  $x_{k-1}$  minimizes  $q(x)$  on  $sp\{q_0, \dots, q_{k-1}\} = K(b, A, k)$ , and this step, which would give  $x_k = x_{k-1}$  in the ORTHODIR algorithm, is skipped in GCD. Thus,  $d_k$  is the first occurring element of  $\{q_{k-1}, \dots, q_t\}$  that is not orthogonal to the gradient of  $q$  at  $x_{k-1}$ .

### The Truncated Generalized Conjugate Directions Algorithm

An alternative to choosing  $d_k$  so that only a fixed number of previous directions are needed in the calculations, while still avoiding the increasing storage and work at each iteration associated with keeping all of the previous directions, is to limit to  $m$  the number of previous directions saved, and at each step to minimize  $q(x)$  over an affine space that is a translate of  $sp\{p_r, \dots, p_{k-1}, d_k\}$ , where  $r = \max(k-m, 1)$ . This results in the truncated version of the Generalized Conjugate Directions algorithm.

#### *Generalized Conjugate Directions (m) Algorithm*

$$x_0 = 0, \bar{r}_0 = h, k = 1.$$

while  $\bar{r}_{k-1} \neq 0$

do

    get  $d_k$  such that  $d_k^T \bar{r}_{k-1} \neq 0$

$$\tau = \max(k-m, 1)$$

$$x_k = \underset{x \in \{x_{\tau-1} + sp\{p_r, \dots, p_{k-1}, d_k\}\}}{\operatorname{argmin}} q(x)$$

$$p_k = x_k - x_{k-1}$$

$$\bar{r}_k = \bar{r}_{k-1} - Hp_k$$

$$k = k + 1$$

end do

Analogous to Theorem 1, we have our second main theorem.

**Theorem 2:** If  $H$  is symmetric positive definite, then the following statements are true about the  $\text{GCD}(m)$  algorithm (Let  $\tau = \max(k-m, 1)$ ):

- (i) The algorithm terminates if and only if  $x_{k-1} = x^*$ .
- (ii)  $x_k$  minimizes  $q(x)$  on  $\{x_{\tau-1} + sp\{p_\tau, \dots, p_k\}\}$ .
- (iii) Every  $p_k$  generated by the algorithm is nonzero.
- (iv)  $\bar{r}_k^T p_j = 0$ ,  $\tau \leq j \leq k$
- (v)  $p_i^T H p_j = 0$ ,  $i \neq j$ ,  $|i-j| \leq m$
- (vi)  $\dim sp\{p_\tau, \dots, p_k\} = k - \tau + 1$
- (vii)  $p_j^T H p_j = p_j^T \bar{r}_{\tau-1}$ ,  $\tau \leq j \leq k$

The proof of Theorem 2 is similar to that of Theorem 1, and is omitted.

The linear system that determines  $x_k$  in the  $\text{GCD}(m)$  algorithm is analogous to that of the GCD algorithm. At each step of the  $\text{GCD}(m)$  algorithm,  $x_k$  is chosen to satisfy

$$x_k = \underset{x \in \{x_{\tau-1} + sp\{p_\tau, \dots, p_{k-1}, d_k\}\}}{\operatorname{argmin}} q(x).$$

Let  $P_k = [p_\tau \dots p_{k-1}, d_k]$ ,  $x = x_{\tau-1} + P_k c$ , and

$$\hat{q}(c) = q(P_k c + x_{\tau-1}) = \frac{1}{2}(P_k c + x_{\tau-1})^T H (P_k c + x_{\tau-1}) - h^T (P_k c + x_{\tau-1}).$$

Then  $\nabla \hat{q}(c) = P_k^T H (P_k c + x_{\tau-1}) - P_k^T h$ .  $P_k$  has full column rank, so that  $P_k^T H P_k$  is positive definite, and  $x_k$  can be found by solving the linear system  $P_k^T H P_k c = P_k^T (h - H x_{\tau-1}) (= P_k^T \bar{r}_{\tau-1})$  for  $c$ , and setting  $x_k = x_{\tau-1} + P_k c$ .

The solution to the linear system is

$$c = \begin{bmatrix} 1 - \alpha_k \frac{p_\tau^T H d_k}{p_\tau^T H p_\tau} \\ \vdots \\ 1 - \alpha_k \frac{p_{k-1}^T H d_k}{p_{k-1}^T H p_{k-1}} \\ \alpha_k \end{bmatrix}, \text{ where } \alpha_k = \frac{d_k^T \bar{r}_{\tau-1} - \sum_{j=\tau}^{k-1} d_k^T H p_j}{d_k^T H d_k - \sum_{j=\tau}^{k-1} \frac{(d_k^T H p_j)^2}{p_j^T H p_j}}.$$

Again, we define for  $\tau \leq j \leq k-1$

$$\beta_j^{(k)} = -\frac{d_k^T H p_j}{p_j^T H p_j}.$$

Then

$$\alpha_k = \frac{d_k^T \bar{r}_{k-1}}{d_k^T H (d_k + \sum_{j=\tau}^{k-1} \beta_j^{(k)} p_j)},$$

and

$$\begin{aligned} x_k &= x_{\tau-1} + P_k c = x_{\tau-1} + \sum_{j=\tau}^{k-1} p_j + \alpha_k (d_k + \sum_{j=\tau}^{k-1} \beta_j^{(k)} p_j) \\ &= x_{k-1} + \alpha_k (d_k + \sum_{j=\tau}^{k-1} \beta_j^{(k)} p_j). \end{aligned}$$

To ensure convergence for the truncated version of the algorithm, we slightly strengthen the requirement that  $d_k$  not be orthogonal to the gradient of  $q$  at  $x_{k-1}$ , asking that the angle between  $d_k$  and  $\bar{r}_{k-1}$  be bounded away from  $90^\circ$ . We use the notation  $\kappa(H) = \lambda_{\max}(H)/\lambda_{\min}(H)$  for the condition number of a symmetric positive definite matrix  $H$ .

The following theorem shows a  $q$ -linear rate of convergence for the entire class of  $\text{GCD}(m)$  algorithms. But it is pessimistic because the rate constant that appears is a lower bound on the amount of reduction that would be achieved with a  $\text{GCD}(0)$  algorithm. We shall remark at the end of the section on the relationship between this bound and other bounds given for special cases.

*Theorem 3:* If  $H$  is symmetric positive definite,  $d_k \neq 0$  for all  $k$ , and there exists  $\gamma > 0$  such that  $|d_k^T \bar{r}_{k-1}| \geq \gamma \|d_k\|_2 \|\bar{r}_{k-1}\|_2$  for all  $k$ , then the sequence of iterates  $\{x_k\}$  generated by a  $\text{GCD}(m)$  algorithm converges to  $x^*$  and satisfies

$$\|x_k - x^*\|_H \leq \left(1 - \frac{\gamma^2}{\kappa(H)}\right)^{\frac{1}{2}} \|x_{k-1} - x^*\|_H.$$

*Proof:*

If  $\bar{r}_{k-1} = 0$  for some  $k$ , then  $x_{k-1} = x^*$ . Assume  $\bar{r}_{k-1} \neq 0$  for all  $k$ . Then

$$\begin{aligned} q(x_k) - q(x_{k-1}) &= \frac{1}{2}(x_{k-1} + p_k)^T H(x_{k-1} + p_k) - h^T(x_{k-1} + p_k) - \frac{1}{2}x_{k-1}^T H x_{k-1} + h^T x_{k-1} \\ &= p_k^T H x_{k-1} + \frac{1}{2} p_k^T H p_k - p_k^T h \\ &= -p_k^T \bar{r}_{k-1} + \frac{1}{2} p_k^T \bar{r}_{k-1} \\ &= -\frac{1}{2} p_k^T \bar{r}_{k-1}, \\ q(x_k) &= \sum_{j=1}^k q(x_j) - q(x_{j-1}) \\ &= -\frac{1}{2} \sum_{j=1}^k p_j^T \bar{r}_{j-1} < 0. \end{aligned}$$

Since  $H$  is positive definite,  $q(x)$  is bounded below, so we conclude that  $\lim_{k \rightarrow \infty} \sum_{j=1}^k p_j^T \bar{r}_{j-1}$

exists. Hence,  $\lim_{k \rightarrow \infty} p_k^T \bar{r}_{k-1} = 0$ , and

$$\begin{aligned} \bar{r}_{k-1}^T p_k &= \bar{r}_{k-1}^T [\alpha_k(d_k + \sum_{j=r}^{k-1} \beta_j^{(k)} p_j)] \\ &= \alpha_k \bar{r}_{k-1}^T d_k \\ &= \frac{(\bar{r}_{k-1}^T d_k)^2}{d_k^T H d_k - \sum_{j=r}^{k-1} \frac{(d_k^T H p_j)^2}{p_j^T H p_j}}. \end{aligned}$$

Since all the terms in the summation are nonnegative, this is

$$\begin{aligned} &\geq \frac{(\bar{r}_{k-1}^T d_k)^2}{d_k^T H d_k} \\ &\geq \frac{\gamma^2 \|\bar{r}_{k-1}\|_2^2 \|d_k\|_2^2}{\lambda_{\max}(H) \|d_k\|_2^2} \\ &= \frac{\gamma^2}{\lambda_{\max}(H)} \|\bar{r}_{k-1}\|_2^2 > 0. \end{aligned}$$

We have  $0 = \lim_{k \rightarrow \infty} \bar{r}_{k-1}^T p_k \geq \frac{\gamma^2}{\lambda_{\max}(H)} \lim_{k \rightarrow \infty} \|\bar{r}_{k-1}\|_2^2 \geq 0$ , so that  $\{\bar{r}_k\}$  converges to 0. Since  $H$

is positive definite, this is necessary and sufficient for  $\lim_{k \rightarrow \infty} x_k = x^*$ .

We have

$$\begin{aligned} q(x_{k-1}) - q(x_k) &= \frac{1}{2} p_k^T \bar{r}_{k-1} \\ &= \frac{1}{2} \alpha_k d_k^T \bar{r}_{k-1} \\ &\geq \frac{\gamma^2}{2\lambda_{\max}(H)} \|\bar{r}_{k-1}\|_2^2, \end{aligned}$$

and

$$\begin{aligned} \|x_k - x^*\|_H^2 &= 2[q(x_k) - q(x^*)] \\ &= 2[q(x_k) - q(x_{k-1}) + q(x_{k-1}) - q(x^*)] \\ &\leq \|x_{k-1} - x^*\|_H^2 - \frac{\gamma^2}{\lambda_{\max}(H)} \|\bar{r}_{k-1}\|_2^2. \end{aligned}$$

Now

$$\bar{r}(x) = h - Hx = H(x^* - x),$$

so that

$$\|x - x^*\|_H^2 = \bar{r}(x)^T H^{-1} \bar{r}(x),$$

and

$$\|\bar{r}_{k-1}\|_2^2 \geq \lambda_{\min}(H) \|x_{k-1} - x^*\|_H^2.$$

Thus,

$$\begin{aligned} \|x_k - x^*\|_H^2 &\leq \|x_{k-1} - x^*\|_H^2 - \frac{\gamma^2 \lambda_{\min}(H)}{\lambda_{\max}(H)} \|x_{k-1} - x^*\|_H^2 \\ &= \left(1 - \frac{\gamma^2}{\kappa(H)}\right) \|x_{k-1} - x^*\|_H^2. \bullet \end{aligned}$$

Convergence of ORTHOMIN( $k$ ) is quickly and easily verified. For ORTHOMIN( $k$ ),  $d_k = r_{k-1}$ , and

$$d_k^T \bar{r}_{k-1} = r_{k-1}^T A^T r_{k-1} = r_{k-1}^T A r_{k-1} = r_{k-1}^T \left(\frac{A + A^T}{2}\right) r_{k-1}.$$

With the assumption that the symmetric part of  $A$  is positive definite, our hypothesis is satisfied.

Let  $M = \frac{A + A^T}{2}$ . Then

$$\begin{aligned} d_k^T \bar{r}_{k-1} &= r_{k-1}^T M r_{k-1} \geq \lambda_{\min}(M) \|r_{k-1}\|_2^2 \\ &= \frac{\lambda_{\min}(M)}{\|A^T\|_2} \|A^T\|_2 \|r_{k-1}\|_2 \|d_k\|_2 \\ &\geq \frac{\lambda_{\min}(M)}{\|A\|_2} \|\bar{r}_{k-1}\|_2 \|d_k\|_2. \end{aligned}$$

If the symmetric part of  $A$  is negative definite, we can use the bound

$$|d_k^T \bar{r}_{k-1}| = |r_{k-1}^T M r_{k-1}| \geq -\lambda_{\max}(M) \|r_{k-1}\|_2^2.$$

If the symmetric part of  $A$  is indefinite, convergence of ORTHOMIN ( $k$ ) is not guaranteed. The algorithm can fail to produce descent at some step in that case, since  $d_k^T \bar{r}_{k-1} = r_{k-1}^T M r_{k-1}$  can be zero without  $r_{k-1}$  being zero.

The bound  $\gamma$  that appears in Theorem 3 is easily obtained for CG, CGNE, Craig's method, and PCG. For CG, CGNE, and Craig's method,  $d_k^T \bar{r}_{k-1} = \|d_k\|_2 \|\bar{r}_{k-1}\|_2$ , so that  $\gamma = 1$ . We have already established a bound for ORTHOMIN ( $k$ ), which also applies to CR, GCR, and ORTHOMIN. Similarly, for PCG one can obtain  $\gamma = 1/\kappa(M^{-1})$ .

A better convergence rate constant than that shown in Theorem 3 can be obtained for particular choices of  $H$  and  $d_k$ . From the proof of Theorem 3

$$q(x_{k-1}) - q(x_k) \geq \frac{(d_k^T \bar{r}_{k-1})^2}{2d_k^T H d_k},$$

which gives

$$\|x_k - x^*\|_H^2 \leq \|x_{k-1} - x^*\|_H^2 - \frac{(d_k^T \bar{r}_{k-1})^2}{d_k^T H d_k},$$

and it may be advantageous to use this rather than the coarser bound involving  $\gamma$  when  $d_k$  and  $H$  are known. For example, for CR, GCR, ORTHOMIN, and ORTHOMIN( $k$ ),

$$H = A^T A, \quad d_k = r_{k-1}, \quad \bar{r}_k = A^T r_k, \quad \text{and} \quad \|x_k - x^*\|_H^2 = \|r_k\|_2^2,$$

so that

$$\frac{(d_k^T \bar{r}_{k-1})^2}{d_k^T H d_k} = \frac{(r_{k-1}^T A r_{k-1})^2}{r_{k-1}^T A^T A r_{k-1}} \geq \frac{\lambda_{\min}^2(M)}{\lambda_{\max}(A^T A)} \|r_{k-1}\|_2^2, \text{ where } M = \frac{A + A^T}{2},$$

and

$$\|r_k\|_2^2 \leq \left(1 - \frac{\lambda_{\min}^2(M)}{\lambda_{\max}(A^T A)}\right) \|r_{k-1}\|_2^2.$$

Elman (1982) derives this q-linear error bound on  $\{\|r_k\|_2\}$  on page 49 in the proof of his Theorem 5.9. However, he weakens it in the statement of the theorem to the r-linear error bound

$$\|r_k\|_2^2 \leq \left(1 - \frac{\lambda_{\min}^2(M)}{\lambda_{\max}(A^T A)}\right)^k \|r_0\|_2^2.$$

For PCG,

$H = A$ ,  $d_k = M^{-1}r_{k-1}$ , where  $M = C^2$  is the preconditioner,  $\bar{r}_k = r_k$ ,

$$\begin{aligned} \frac{(d_k^T \bar{r}_{k-1})^2}{d_k^T H d_k} &= \frac{(r_{k-1}^T M^{-1} r_{k-1})^2}{r_{k-1}^T M^{-1} A M^{-1} r_{k-1}} = \frac{[(C^{-1} r_{k-1})^T (C^{-1} r_{k-1})]^2}{(C^{-1} r_{k-1})^T (C^{-1} A C^{-1}) (C^{-1} r_{k-1})} \geq \frac{\|C^{-1} r_{k-1}\|_2^2}{\lambda_{\max}(C^{-1} A C^{-1})} \\ &\geq \frac{\lambda_{\min}(C^{-1} A C^{-1})}{\lambda_{\max}(C^{-1} A C^{-1})} \|C^{-1} r_{k-1}\|_{CA^{-1}C}^2 = \frac{\|x_{k-1} - x^*\|_A^2}{\kappa(C^{-1} A C^{-1})}, \end{aligned}$$

and

$$\|x_k - x^*\|_A^2 \leq \left(1 - \frac{1}{\kappa(C^{-1} A C^{-1})}\right) \|x_{k-1} - x^*\|_A^2.$$

This q-linear error bound implies the weaker r-linear error bound

$$\|x_k - x^*\|_A^2 \leq \left(1 - \frac{1}{\kappa(C^{-1} A C^{-1})}\right)^k \|x_0 - x^*\|_A^2.$$

The Chebychev polynomial approach yields only an r-linear error bound, but it is much better than the r-linear error bound above, since  $\kappa(C^{-1} A C^{-1})^{\frac{1}{2}}$  appears in place of  $\kappa(C^{-1} A C^{-1})$ .

### Conjugate Directions in Quadratic Programming

The formulation of the generalized conjugate directions algorithm that has been presented for solving systems of linear equations can be readily adapted to give a conjugate directions algorithm for solving the equality constrained quadratic programming problem



$$\begin{aligned} & \text{minimize} && q(x) = \frac{1}{2} x^T H x - h^T x \\ & \text{subject to} && Ax = 0, \end{aligned}$$

where we assume that  $H$  is symmetric positive definite on the nullspace of  $A$ , and that  $A$  has full row rank. Given an initial feasible point, a problem with inhomogeneous constraints can readily be put in this form.

Starting with the initial feasible point  $x_0=0$ , the algorithm will maintain feasibility at each step, and will produce a collection of linearly independent, mutually  $H$ -conjugate direction vectors  $\{p_i\}$  as in the GCD algorithm. Here, the direction vectors will all be in the nullspace of  $A$ , and the algorithm will terminate after no more than  $s$  steps, where  $s$  is the dimension of the nullspace of  $A$ . The case where  $A$  is the row of all ones arises in some problems from conservation laws.

Let  $P$  denote the projector onto the nullspace of  $A$ :  $P = I - A^T(AA^T)^{-1}A$ . To solve the quadratic programming problem, the GCD algorithm is modified so that the termination criterion is that the projected gradient,  $P\bar{r}_{k-1}$ , is zero, and at each step we

choose  $d_k$  so that  $d_k$  is not orthogonal to the projected gradient of  $q$  at the current point:

$$d_k^T P\bar{r}_{k-1} \neq 0;$$

let  $\hat{d}_k = Pd_k$ ;

minimize  $q(x)$  over  $sp\{p_1, \dots, p_{k-1}, \hat{d}_k\}$ .

Since  $P$  is symmetric,  $d_k^T P\bar{r}_{k-1} = \hat{d}_k^T \bar{r}_{k-1}$ , so that the modified algorithm operates in exactly the same way as the GCD algorithm, and Theorem 1, with  $n$  in part (i) changed to  $s$ , applies to the quadratic programming conjugate directions algorithm.

### Acknowledgment

The authors wish to thank the referee, whose helpful comments improved this presentation.

## REFERENCES

- AXELSSON, O. [1980]. Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations, *Linear Algebra Appl.*, 29, pp. 1-16.
- CANTRELL, J.W. [1969]. Relation between the memory gradient method and the Fletcher-Reeves method, *JOTA*, vol. 4, pp. 67-71
- CRAGG, E.E. and A.V. LEVY [1969]. Study on a supermemory gradient method for the minimization of functions, *JOTA* vol. 4, pp. 191-205
- EISENSTAT, S.C., H.C. ELMAN and M.H. SCHULTZ [1983]. Variational iterative methods for non-symmetric systems of linear equations, *SIAM J. Numer. Anal.*, vol. 20, pp. 345-357.
- ELMAN, H.C. [1982]. Iterative methods for large, sparse, nonsymmetric systems of linear equations, Ph.D. Thesis, Yale University. Also available as Research Report #229.
- GOLUB, G.H. and C.F. VAN LOAN [1983]. *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland.
- MIELE, A. and J.W. CANTRELL [1969]. Study on a memory gradient method for the minimization of functions, *JOTA* vol. 3, pp. 459-470
- NAZARETH, L. [1984]. Some approaches to making conjugate gradient methods for nonlinear minimization less dependent on notions of conjugacy, University of California, Berkeley, Center for Pure and Applied Mathematics.
- SAAD, Y. and M.H. SCHULTZ, [1983]. A Generalized Minimal Residual algorithm for solving nonsymmetric linear systems, Yale University, Department of Computer Science, Research Report RR-254.
- SAAD, Y. and M.H. SCHULTZ, [1985]. Conjugate gradient-like algorithms for solving nonsymmetric linear systems, *Math Comp.*, vol. 44, pp. 417-424.
- VINSOME, P.K.W. [1976]. Orthomin, an iterative method for solving sparse sets of simultaneous linear equations, in Society of Petroleum Engineers of AIME, *Proceedings of the Fourth Symposium on Reservoir Simulation*.
- YOUNG, D.M. and K.C. JEA [1980]. Generalized conjugate gradient acceleration of iterative methods. *Linear Algebra Appl.*, vol. 34, pp. 159-194.
- YOUNG, D.M., L.J. HAYES and K.C. JEA [1981]. Generalized Conjugate gradient acceleration of iterative methods part I: The symmetrizable case, Report #162, Center for Numerical Analysis, The University of Texas at Austin.
- YOUNG, D.M. and K.C. JEA [1981]. Generalized conjugate gradient acceleration of iterative methods part II: The nonsymmetrizable case, Report #163, Center for Numerical Analysis, The University of Texas at Austin.